

IN THE CLAIMS

1. (Currently Amended) A method, operable by a processor, for changing a first scheduler in a virtual machine monitor, comprising:

loading a second scheduler in the virtual machine monitor when the virtual machine monitor is running; and

activating the loaded second scheduler to handle a scheduling request for a scheduling process in place of the first scheduler when the virtual machine monitor is running, wherein an address, corresponding to a function in a function pointer array and associated with the first scheduler, is to be dynamically patched into the scheduling request to allow the scheduling request to be sent to the [[first]] second scheduler directly,

wherein the loading further comprises ceasing device resources owned by a running virtual machine in response to receiving a scheduler changing request to change the first scheduler and loading the second scheduler in the virtual machine monitor based upon a scheduler parameter of the scheduler changing request, wherein the loading further comprises unloading the first scheduler from the virtual machine monitor before loading the second scheduler.

2-3 (Canceled)

4. (Original) The method of claim 1, wherein the activating further comprises:
replacing a first scheduler identifier with a second scheduler identifier to route between the second scheduler and a requester that generated the scheduling request, when the virtual machine monitor is running.

5. (Original) The method of claim 1, wherein the activating further comprises:
replacing a first function pointer array pointing to a first function array of the first scheduler with a second function pointer array pointing to a second function array of the second scheduler to route between the second scheduler and a requester that generated the request, when the virtual machine monitor is running.

6. (Original) The method of claim 1, wherein the activating further comprises:
dynamically patching an address associated with the second scheduler into the scheduling request when the virtual machine monitor is running.

7. (Original) The method of claim 1, further comprising:

unloading the second scheduler from the virtual machine monitor when the virtual machine monitor is running; and

re-activating the first scheduler to handle a scheduling request after the second scheduler has been unloaded.

8. (Currently Amended) A virtual machine monitor for changing a first scheduler, comprising:

a loading logic hardware to load a second scheduler in the virtual machine monitor when the virtual machine monitor is running; and

an activating logic to activate the loaded second scheduler to handle a scheduling request for a scheduling process in place of the first scheduler when the virtual machine monitor is running, wherein an address, corresponding to a function in a function pointer array and associated with the first scheduler, is to be dynamically patched into the scheduling request to allow the scheduling request to be sent to the [[first]] second scheduler directly,

wherein the loading logic is further to cease device resources owned by a running virtual machine in response to receiving a scheduler changing request to change the first scheduler and load the second scheduler in the virtual machine monitor based upon a scheduler parameter of the scheduler changing request, wherein the loading logic is further to unload the first scheduler from the virtual machine monitor before loading the second scheduler.

9-10. (Canceled)

11. (Original) The virtual machine monitor of claim 8, wherein the activating logic is further to:

replace a first scheduler identifier with a second scheduler identifier;

route between the second scheduler as identified by the second scheduler identifier and a requester that generated the scheduling request, when the virtual machine monitor is running.

12. (Original) The virtual machine monitor of claim 8, wherein the activating logic is further to:

replace a first function pointer array pointing to a first function array of the first scheduler with a second function pointer array pointing to a second function array of the second scheduler;

route between the second function array pointed by the second function pointer array and a requester that generated the scheduling request, when the virtual machine monitor is running.

13. (Original) The virtual machine monitor of claim 8, wherein the activating logic is further to:

 dynamically patch an address associated with the second scheduler into a scheduling request when the virtual machine monitor is running.

14. (Original) The virtual machine monitor of claim 8, wherein the loading logic is further to unload the second scheduler from the virtual machine monitor when the virtual machine monitor is running; and the activating logic is further to re-activate the first scheduler to handle a scheduling request after the second scheduler has been unloaded.

15-23. (Canceled)

24. (Currently Amended) A non-transitory computer readable medium comprising a plurality of instructions that in response to being executed result in an apparatus:

 loading a second scheduler in a virtual machine monitor when the virtual machine monitor is running; and

 activating the loaded second scheduler to handle a scheduling request for a scheduling process in place of a first scheduler, when the virtual machine monitor is running, wherein an address, corresponding to a function in a function pointer array and associated with the first scheduler, is to be dynamically patched into the scheduling request to allow the scheduling request to be sent to the [[first]] second scheduler directly,

wherein the plurality of instructions that result in the apparatus loading the second scheduler, further result in the apparatus: ceasing device resources owned by a running virtual machine in response to receiving a scheduler changing request to change the first scheduler; and loading the second scheduler in the virtual machine monitor based upon a scheduler parameter of the scheduler changing request, wherein the plurality of instructions further result in the apparatus unloading the first scheduler from the virtual machine monitor before the second scheduler is loaded.

25-26. (Canceled)

27. (Previously Presented) The computer readable medium of claim 24, wherein the plurality of instructions that result in the apparatus activating the second scheduler, further result in the apparatus:

replacing a first scheduler identifier with a second scheduler identifier to route between the second scheduler and a requester that generated the scheduling request, when the virtual machine monitor is running.

28. (Previously Presented) The computer readable medium of claim 24, wherein the plurality of instructions that result in the apparatus activating the second scheduler, further result in the apparatus:

replacing a first function pointer array pointing to a first function array of the first scheduler with a second function pointer array pointing to a second function array of the second scheduler to route between the second scheduler and a requester that generated the scheduling request, when the virtual machine monitor is running.

29. (Previously Presented) The computer readable medium of claim 24, wherein the plurality of instructions that result in the apparatus activating the second scheduler, further result in the apparatus:

dynamically patching an address associated with the second scheduler to the scheduling request when the virtual machine monitor is running.

30. (Previously Presented) The computer readable medium of claim 24 wherein the plurality of instructions further result in the apparatus:

unloading the second scheduler from the virtual machine monitor when the virtual machine monitor is running; and

re-activating the first scheduler to handle the scheduling request after the second scheduler has been unloaded.

31. (Previously Presented) The method of claim 1, wherein the virtual machine monitor is to comprise the function pointer array.

32. (New) The method of claim 1, wherein the addresses associated with the first scheduler are to be dynamically patched into the scheduling request during a scheduler model loading or unloading stage.